Large Scale Computer Algebra Calculations In Field Theory

J.A.M. Vermaseren

Nikhef

- Introduction
- Mincer algorithms
- Structure functions and Harmonic sums
- Multiple Zeta Values and Euler sums
- Relations
- The program
- Results $\longrightarrow$ basis?
- Conclusions

Based on work done with J. Blümlein and D. Broadhurst.

Introduction

When we do calculations in Quantum Field Theory we have several options:

- Work non-perturbatively.
- Use a method that isn't based on Feynman diagrams.
- Use Feynman diagrams.

In this talk we concentrate on the Feynman diagrams.

Here we have various categories, each of which is treated differently.

- All particles are massless and we have at most one loop. In this case there are new methods (MHV) that can give impressive results, even when the number of external legs becomes very large.

- Some or all particles have a mass and the number of loops is at most one. This is a focal point of research when looking at the effects of the weakly interacting particles, or when looking at heavy quark production.

- We have more loops and maybe one or two mass scales, as in $g-2$ calculations. The number of external legs is very limited.

- We have a large number of loops, but few external legs and no masses.

Here we will look at the last case. In this case the number of diagrams can become very large. No methods exist yet to deal with clusters of diagrams as a whole and hence the only way to deal with such calculations is by using highly automated computer programs that process diagram by diagram.

(We ignore here the possibility to add all diagrams of the same topology before processing and then deal with them in one program).

This again requires very special techniques because we cannot assume that the computer knows 'smart tricks'.

What is needed is a fully automated set of algoritms that will work out any diagram of the given category. This makes it nearly impossible to use certain types of relations. Examples:

Easy:

$$\frac{1}{(a+x)}\frac{1}{(b+x)} \to (\frac{1}{(a+x)} - \frac{1}{(b+x)})\frac{1}{(b-a)}, \quad a \neq b$$

and one can add delta and theta functions when there is the risk that $a = b$.

Nearly impossible:

$$\epsilon^{\mu\nu\rho\sigma}\delta^{\kappa\lambda} - \epsilon^{\kappa\nu\rho\sigma}\delta^{\mu\lambda} - \epsilon^{\mu\kappa\rho\sigma}\delta^{\nu\lambda} - \epsilon^{\mu\nu\kappa\sigma}\delta^{\rho\lambda} - \epsilon^{\mu\nu\rho\kappa}\delta^{\sigma\lambda} \to 0$$

Hence one is confronted with a certain handicap from the smart side, while at the same time there is an enormous advantage with respect to speed and chance of making errors. The selection of the algorithms needs to take this into account.

One has to be stupid in a smart way.

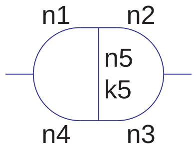
Mincer algorithms

The basic algorithm for two and three loop propagator diagrams with only massless lines is integration by parts. We start with

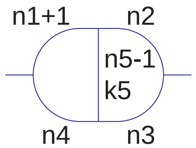
$$\int d^D P \frac{\partial}{\partial P_\mu} \left(R^\mu I(n_1, \cdots, n_m) \right) = 0$$

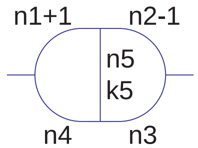
in which P is one of the loop momenta, I is a collection of propagators and dotproducts with n_i the powers of those objects, and R is any momentum, preferably one of the loop momenta or the external momentum.
Working out the derivative gives useful identities in which we obtain integrals with different values for the n_i.

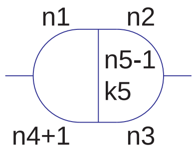
This is illustrated in the following graphical equation:

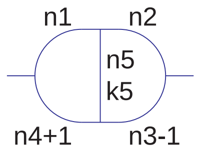


$$= \frac{1}{D+k_5-2n_5-n_1-n_4}\Big($$

$+n_1$ 

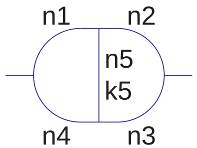
$-\, n_1$ 

$+n_4$ 

$-\, n_4$ 

$\Big)$

in which we have defined



$$= \int d^D p_1 \; d^D p_2 \frac{S{\cdot}p_5^{k_5}}{(p_1^2)^{n_1}(p_2^2)^{n_2}(p_3^2)^{n_3}(p_4^2)^{n_4}(p_5^2)^{n_5}}$$

and used $P \longrightarrow p_5$, $R \longrightarrow p_5$ and S is any external momentum.

We see that we can eventually make one of the parameters n_2, n_3, n_5 to be zero, thereby removing one line and creating simpler integrals.

Using the IBP techniques we can reduce all two and three loop propagator graphs to just three types of integrals. These integrals are known to sufficient powers in $\epsilon = (4 - D)/2$.

The massless propagator graphs lead to dimensionless answers (apart from overal factors).

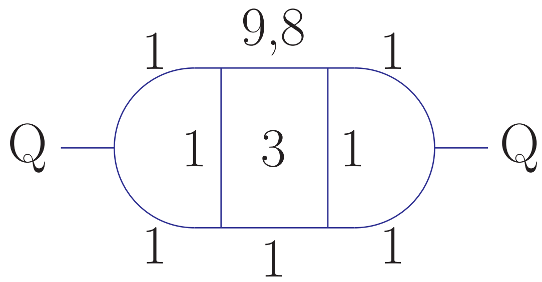
Structure functions and Harmonic sums

For structure functions one has a single parameter, usually Bjorken x. The Q^2 dependence is in the coupling constant and hence plays no direct role in the evaluation of the Feynman diagrams. The extra parameter makes the integrals much more complicated. One way to do them is by making a Mellin transform:

$$f(N) = \int_0^1 dx\ x^{N-1} f(x)$$

Now the variable is the (integer) N.

The first observation is that if we give N a value, like 2 or 4, we can convert the calculation to a (more complicated) propagator calculation as in (assuming that $P{\cdot}P = 0$):



$$= \int d^D p_1 d_2^D d^D p_3 \frac{(2P{\cdot}p_2)^8 (2p_7{\cdot}p_8)^3}{(p_1^2)(p_2^2)^9(p_3^2)(p_4^2)(p_5^2)(p_6^2)(p_7^2)(p_8^2)}$$

$$= (Q^2)^{1-6\epsilon}(\frac{2P{\cdot}Q}{Q^2})^8 \times$$

$$(\frac{1}{56}\epsilon^{-2} + \frac{107}{3360}\epsilon^{-1} - \frac{39}{35}\zeta_3 + \frac{436990847}{326592000})$$

This took 1.87 sec. with TFORM on a quad Xeon computer at 3.2 GHz. (127 sec. with (an older version of) FORM on a P1000 computer.)

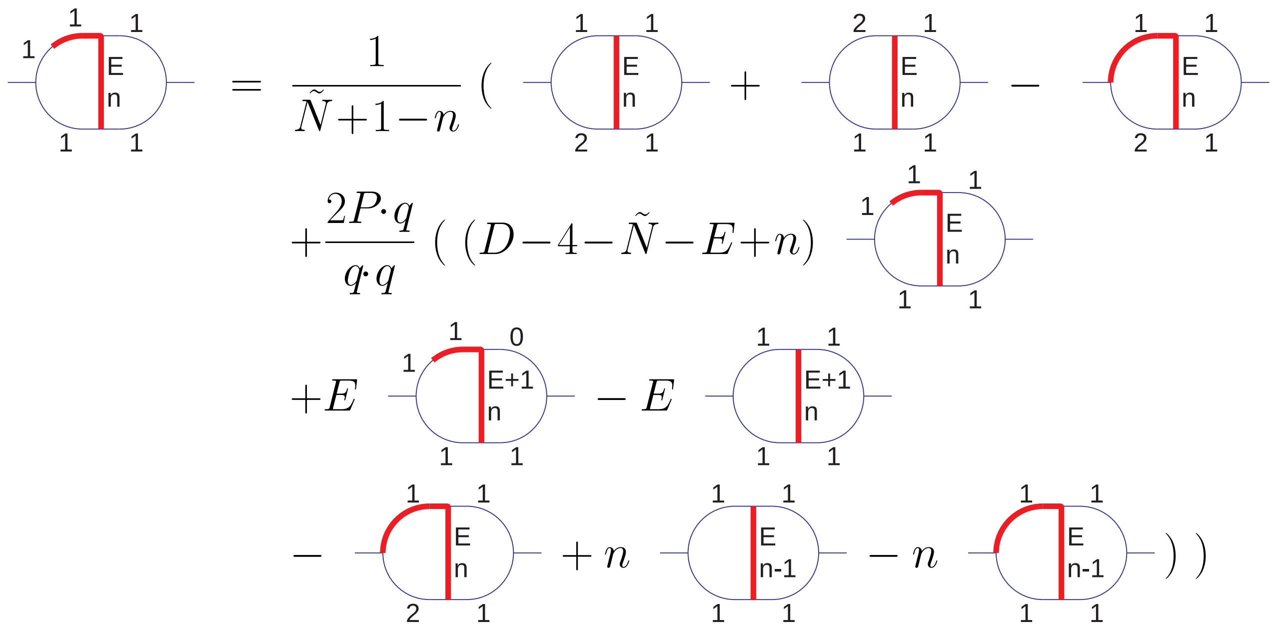
```
    #define TOPO "la"
    #define SCHEME "1"
    #include- mincer.h
    off statistics;
    .global
    Local F = Q.Q^-1*(2*p7.p8)^3*(2*P.p2)^8/p1.p1/p2.p2^9/p3.p3/p4.p4
              /p5.p5/p6.p6/p7.p7/p8.p8;
    #call harmo(P,Q,mncFFPP)
    .sort
    Multiply ep^3;
    #call integral(`TOPO')
~~~Answer in the G-scheme
    .sort
    Multiply 1/2^8;
    Print +s;
    .end

   F =
       + 436990847/326592000
       + 1/56*ep^-2
       + 107/3360*ep^-1
       - 39/35*z3
      ;

  0.18 sec + 6.49 sec: 6.67 sec out of 1.87 sec
```

This means that the program Mincer, written for the massless propagators can be used for calculating a number of Mellin moments. Because for each moment the calculation becomes more time consuming by a factor this has only been done up to about $N = 16$. These moments can then be used to make a partial reconstruction of the function of Bjorken x. Such a reconstruction is typically good for values $x \geq 1/N_{max}$.

To get the complete reconstruction one needs an infinite number of even or odd values of N. Hence we need basically the complete function of N. This requires a much more complicated scheme than the Mincer scheme that involves more equations than just the integration by parts. One such relation is:



What we see is that with more variables there is more room for artistic expression. The red line is the P-flow of the parton through the diagram. After about N derivatives it it is set to zero.

(Formula from a paper by S.O.Moch and J.V.)

One needs to introduce also a category of functions that will be large enough to contain the answers. These functions are the Harmonic sums. They are defined by:

$$\begin{aligned}
S_m(N) &= \sum_{i=1}^{N} \frac{1}{i^m} \\
S_{-m}(N) &= \sum_{i=1}^{N} \frac{(-1)^i}{i^m} \\
S_{m,m_2,\cdots,m_p}(N) &= \sum_{i=1}^{N} \frac{1}{i^m} S_{m_2,\cdots,m_p}(i) \\
S_{-m,m_2,\cdots,m_p}(N) &= \sum_{i=1}^{N} \frac{(-1)^m}{i^m} S_{m_2,\cdots,m_p}(i)
\end{aligned}$$

This is a notation that is also suitable for computers. There is a difference here between various definitions as there are also people using $i-1$ for the argument of the S in the recursive formula. Those sums we call Z-sums.

Eventually a scheme can be devised to break down all three loop graphs either to simpler integrals or to integrals determined by a difference equation. With the ansatz that the solution of these equations is a linear combination of Harmonic sums, nearly all of them can be converted into a large set of linear equations which can be solved.
There is one object that is not in this category but it can be taken separately (it is rather rare) and cancels in the end between the diagrams.
The above method is extremely time consuming, both in designing the algorithms and in the running on the computer. This was improved considerably by tabulating intermediate results. In the final runs there were about 3 Gbytes of tabulated integrals. Special features for this were invented for FORM to deal with this. These are called the tablebases.

Multiple Zeta Values and Euler sums

When the results of the calculations in Mellin N-space have to be converted to results in Bjorken x-space one has to perform a so-called inverse Mellin transformation. The functions we will obtain in x-space had to be defined as well. They are called Harmonic polylogarithms. Their definition is:

The harmonic polylogarithms are defined by:

$$
\begin{aligned}
H(0;x) &= \ln x \\
H(1;x) &= \int_0^x \frac{dx'}{1-x'} = -\ln(1-x) \\
H(-1;x) &= \int_0^x \frac{dx'}{1+x'} = \ln(1+x)
\end{aligned}
$$

and the functions

$f(0;x) = \frac{1}{x}, \quad f(1;x) = \frac{1}{1-x}, \quad f(-1;x) = \frac{1}{1+x}$

If $\vec{a}_w$ is an array with w elements, all with value a, then:

$$
\begin{aligned}
H(\vec{0}_w;x) &= \frac{1}{w!} \ln^w x \\
H(a,\vec{m}_w;x) &= \int_0^x dx' \ f(a;x') \ H(\vec{m}_w;x')
\end{aligned}
$$

The harmonic polylogarithms form a 'shuffle' algebra as in

$$\begin{aligned} H_{a,b}(x)H_{c,d}(x) &= H_{a,b,c,d}(x) + H_{a,c,b,d}(x) + H_{a,c,d,b}(x) \\ &\quad + H_{c,a,b,d}(x) + H_{c,a,d,b}(x) + H_{c,d,a,b}(x) \end{aligned}$$

The harmonic sums form a 'stuffle' algebra which is based on properties of sums:

$$\begin{aligned} S_{a,b}(N)S_{c,d}(N) &= S_{a,b,c,d}(N) + S_{a,c,b,d}(N) + S_{a,c,d,b}(N) \\ &\quad + S_{c,a,b,d}(N) + S_{c,a,d,b}(N) + S_{c,d,a,b}(N) \\ &\quad - S_{a\&c,b,d}(N) - S_{a,c\&b,d}(N) - S_{a,c,b\&d}(N) \\ &\quad - S_{c,a,b\&d}(N) - S_{c,a\&d,b}(N) + S_{a\&c,b\&d}(N) \end{aligned}$$

For the Z-sums the minus signs should be replaced by plus signs.

The stuffle addition & is defined by

$$a\&b = \sigma_a \sigma_b (|a| + |b|)$$

with σ_a being the sign of a and σ_b being the sign of b.

We can define a unified notation as in:

$$\begin{aligned} H_{0,0,1,0,-1} &= H_{3,-2} \\ S_{7,-2,1} &= S_{0,0,0,0,0,0,1,0,-1,1} \end{aligned}$$

The notation with the $0, 1, -1$ we call integral notation and the other notation we call sum notation. The number of indices in the integral notation is the weight, and the number of indices in the sum notation is the depth.

For non-alternating sums we have $Z_{\vec{p}}(\infty) = H_{\vec{p}}(1)$ With alternating sums there can be signs. Trivially programmable.

The integral notation allows us to see how many of these sums and integrals exist. There are $2\ 3^{w-1}$ sums and 3^w integrals for weight w. ($2^{w-1}, 2^w$ when we exclude the negative indices).

For the inverse Mellin transform of the three loop structure functions we have to invert weight 6 sums. Let us have a look at one of them ($S_{-1,3,-2}(N)$):

```
#define SIZE "6"
#include- harmpol.h
Off statistics;
.global
Local F = S(R(-1,3,-2),N);
#call invmel(S,N,H,x)
Print +f +s;
.end
```

```
F =
    - sign_(N)*H(R(1,0,0),x)*Htab2(0,-1)*[1+x]^-1
    - sign_(N)*Htab5(0,-1,0,0,-1)*[1+x]^-1
    - sign_(N)*Htab5(0,-1,0,0,1)*[1+x]^-1
    + sign_(N)*Htab5(0,-1,1,0,0)*[1+x]^-1
    - 2*sign_(N)*Htab5(0,0,-1,0,1)*[1+x]^-1
    - 3*sign_(N)*Htab5(0,0,0,-1,1)*[1+x]^-1
    - 3*sign_(N)*Htab5(0,0,0,1,-1)*[1+x]^-1
```

```
- sign_(N)*Htab5(0,0,1,0,-1)*[1+x]^-1
+ sign_(N)*Htab5(0,1,-1,0,0)*[1+x]^-1
+ sign_(N)*Htab5(0,1,0,-1,0)*[1+x]^-1
+ sign_(N)*Htab5(0,1,0,0,-1)*[1+x]^-1
+ sign_(N)*Htab5(1,0,-1,0,0)*[1+x]^-1
+ 2*sign_(N)*Htab5(1,0,0,-1,0)*[1+x]^-1
+ 3*sign_(N)*Htab5(1,0,0,0,-1)*[1+x]^-1
- H(R(-1),x)*Htab4(0,0,-1,0)*[1-x]^-1
+ H(R(-1,-3,0),x)*[1-x]^-1
- H(R(-1,0),x)*Htab3(0,-1,0)*[1-x]^-1
- H(R(-1,0,0),x)*Htab2(-1,0)*[1-x]^-1
+ 6*Htab5(-1,-1,0,0,0)*[1-x]^-1
+ 5*Htab5(-1,0,-1,0,0)*[1-x]^-1
+ 3*Htab5(-1,0,0,-1,0)*[1-x]^-1
+ 4*Htab5(0,-1,-1,0,0)*[1-x]^-1
+ 3*Htab5(0,-1,0,-1,0)*[1-x]^-1
+ 2*Htab5(0,0,-1,-1,0)*[1-x]^-1
+ Htab5(0,0,-1,0,-1)*[1-x]^-1
+ Htab6(-1,0,-1,0,0,-1)
+ Htab6(-1,0,-1,0,0,1)
+ 2*Htab6(-1,0,0,-1,0,1)
+ 3*Htab6(-1,0,0,0,-1,1)
+ 3*Htab6(-1,0,0,0,1,-1)
```

```
 + Htab6(-1,0,0,1,0,-1)
 + 2*Htab6(0,-1,-1,0,0,-1)
 + 2*Htab6(0,-1,-1,0,0,1)
 + Htab6(0,-1,0,-1,0,-1)
 + 3*Htab6(0,-1,0,-1,0,1)
 + 2*Htab6(0,-1,0,0,-1,-1)
 + 5*Htab6(0,-1,0,0,-1,1)
 + 3*Htab6(0,-1,0,0,1,-1)
 + Htab6(0,-1,0,1,0,-1)
 + 4*Htab6(0,0,-1,-1,0,1)
 + 5*Htab6(0,0,-1,0,-1,1)
 + 3*Htab6(0,0,-1,0,1,-1)
 + Htab6(0,0,-1,1,0,-1)
 + 6*Htab6(0,0,0,-1,-1,1)
 + 3*Htab6(0,0,0,-1,1,-1)
;
```

The Htab objects are Hpl's in one in which for instance Htab6(0,0,0,-1, 1,-1) stands for $H_{-4,1,-1}(1)$ and they are related to the sums in infinity.

What we notice is that there are many terms in which we have the sums in infinity or the Hpl's in one. As Euler knew already, there are relations between the sums in infinity as in:

$$
\begin{aligned}
\zeta_6 &= \frac{8}{35}\zeta_2^3 \\
\zeta_{5,1} &= \frac{6}{35}\zeta_2^3 - \frac{1}{2}\zeta_3^2 \\
\zeta_{4,2} &= -\frac{32}{105}\zeta_2^3 + \zeta_3^2 \\
\zeta_{4,1,1} &= \frac{23}{70}\zeta_2^3 - \zeta_3^2 \\
\zeta_{3,3} &= -\frac{4}{35}\zeta_2^3 + \frac{1}{2}\zeta_3^2 \\
\zeta_{3,2,1} &= -\frac{29}{30}\zeta_2^3 + 3\zeta_3^2 \\
\zeta_{3,1,2} &= \frac{53}{105}\zeta_2^3 - \frac{3}{2}\zeta_3^2 \\
\zeta_{2,4} &= \frac{10}{21}\zeta_2^3 - \zeta_3^2 \\
\zeta_{2,2,2} &= \frac{3}{70}\zeta_2^3 \\
\zeta_{2,1,3} &= -\frac{13}{70}\zeta_2^3 + \zeta_3^2
\end{aligned}
$$

These come from the algebra relations for the sums and the polylogarithms taken in infinity, resp one. These are different relations and hence they can be combined to express all these Euler sums in terms of a minimal set. When this is done the answer to the inverse Mellin problem we showed above becomes:

```
 #define SIZE "6"
 #include- harmpol.h
 Off statistics;
 .global
 Local F = S(R(-1,3,-2),N);
 #call invmel(S,N,H,x)
 Print +f +s;
 .end

F =
    - 51/32*[1-x]^-1*z5
    + 3/4*[1-x]^-1*z2*z3
    - 7/2*s6
    + 51/32*z5*ln2
    - 33/64*z3^2
    + 9/4*z2*z3*ln2
    + 121/840*z2^3
    - 51/32*sign_(N)*[1+x]^-1*z5
    + 3/4*sign_(N)*[1+x]^-1*z2*z3
    - 1/2*sign_(N)*H(R(1,0,0),x)*[1+x]^-1*z2
    + 21/20*H(R(-1),x)*[1-x]^-1*z2^2
    + H(R(-1,-3,0),x)*[1-x]^-1
    + 3/2*H(R(-1,0),x)*[1-x]^-1*z3
    + 1/2*H(R(-1,0,0),x)*[1-x]^-1*z2
   ;
```

This is much simpler and much more informative.

In order to be able to give meaningful answers to the calculation of objects like beta functions and anomalous dimensions we need to know the relations between these constants. In its generality this turns out to be a formidable mathematical problem that is as of now unsolved.

In the rest of this talk we are going to see how far we can get with it.

Terminology:

The sums that do not contain the $(-1)^i$ are called non-alternating sums or Multiple Zeta Values (MZVs). The sums that involve $(-1)^i$ are called alternating or Euler sums.

In general one can use even higher roots of unity than just the second root. According to Broadhurst Feynman diagrams can eventually involve up to the sixth root of unity. Very little is known about these functions.

Relations

Unfortunately there is no known constructive way to take one of these constants and express it into a basis. Already there are problems in determining what constitutes a good basis.
The only two ways to express them in an independent set that are currently known are:

- Write down all algebraic relations for these objects and solve the system of equations. Then tabulate all MZVs (Euler sums) and use table substitution afterwards.
- Guess a relation and fit the coefficients with a program like PSLQ after computing all objects in the relation numerically to a very large number of digits. Broadhurst has done much of this in the 1990's.

For the MZVs the only algebraic relations that are known are the shuffle and the stuffle relations we saw before. On does have to include the relations that involve divergent sums and integrals. This divergence is however very mild (logarithmic) and can easily be regularized.
For the Euler sums more relations exist that add nontrivial information. Let us start with:

$$S_{n_1,\cdots,n_p}(N) \;=\; 2^{n_1+\cdots+n_p-p}\sum_{\pm} S_{\pm n_1,\cdots,\pm n_p}(2N)$$

These are called the doubling relations. When $n \to \infty$ and the sums are finite, this gives useful relations.
For the Euler sums there is yet another category of relations which we call the generalized doubling relations (GDR's). They are based on similar principles but we have only a computer algorithm to generate them. No closed formula.

The doubling relations are necessary for weights of 8 and higher. The generalized doubling relations are needed for weight 11 and higher. They have another useful property which we see already at weight 6:

$$Z_{-4,-2} \;=\; -H_{-4,2} = \frac{97}{420}\zeta_2^3 - \frac{3}{4}\zeta_3^2$$

To derive this equation with shuffles and stuffles alone the number of relations one needs is enormous:

depth	shuffles	stuffles
2	11	8
3	52	19
4	72	41

Using the GDR's one needs only relations involving depth 2 (or lower) objects (and there are only 6 GDR's at depth 2, weight 6).

With the GDR's we can calculate relations for objects up to a given depth without needing equations for greater depth. This will allow us to obtain some interesting relations.

We have not seen any violations of this rule.

It is possible to use the stuffle relations to express the divergent sums in terms of powers of the basic divergence $(S_1(\infty))$ and finite sums of lower weights. All divergent sums have the first index being one. Hence we need to compute only $4\ 3^{w-2}$ Euler sums. In the case of the MZVs there is an additional duality relation and the number of MZVs that we need to determine is 2^{w-3}.

For the Euler sums a reasonable basis has been conjectured by Broadhurst. It is made from all sums of a given weight that have negative odd indices and the string of the indices form a Lyndon word.
A Lyndon word is a string of 'letters' that is uniquely minimal when all its cyclic permutations are considered.
Example:

$$H_{-7} \quad H_{-5,-1,-1} \quad H_{-3,-3,-1} \quad H_{-3,-1,-1,-1,-1}$$

This assumes that we express all sums in terms of products of lower weight objects and as few objects of the same weight as possible.

For the MZVs the situation is that there is a conjectured basis made from all Lyndon words that consist of 2's and 3's (which add up to the weight). This turns out to be a very unpleasant basis from the viewpoint of computations.
We prefer a basis in which the depth of the elements is minimal. In that case it seems that it suffices to consider only relations up to $W/3$ to determine the basis.
Our computer programs automatically construct such a basis, but it is far from unique.
We have found however one class of bases with very nice properties. We will see it in due time.

The program

Trying to solve large systems of equations can be quite a challenge. And because we want to reach the limits of what is possible we need the most powerful program we can lay our hands on. Of course we have FORM, but there is also the newer TFORM that can make use of multi-core machines. This gives added power.
We use the MZV program as a test under extreme conditions for TFORM. This has enabled us to

- Test and improve TFORM.
- Improve the program for solving sets of equations.
- Get more results on MZVs (Euler sums).

The first thing to consider is that it may not be possible to have all equations in memory simultaneously. Hence we should select a method that doesn't need this.

So how do we solve 5×10^6 equations with 2×10^6 unknowns?

We would like to go beyond what M. Kaneko, M. Noro and K. Tsurumaki managed. They treated this as a matrix problem (with a size of $2^{W-3} \times 2^{W-4}$) and went to W=20. Using calculus modulus a 15 bits prime they needed about 18 Gbytes of memory and could not go beyond this.

W	size	time
16	72M	150
17	288M	880
18	1.2G	5000
19	4.6G	33000
20	18G	245000

Parameters of the Kaneko et al program on an 8 core computer.

The program managed to determine the size of a basis. The size was according to the Zagier conjecture ($N(W) = N(W-2) + N(W-3)$). It should be noted that the matrix is sparse. In our program the weight 20 expression has at its worst 4158478 terms (100 Mbytes) which means that only one in 2000 entries of the matrix would not be zero.

We start generating a master expression which contains one term for each sum that we want to compute. For the MZVs of weight 4 this expression looks in computer terms like

```
FF =
   +E(0,0,0,1)*(H(0,0,0,1))
   +E(0,0,1,1)*(H(0,0,1,1))
   +E(0,1,0,1)*(H(0,1,0,1));
```

We have used already that we will only compute the finite elements and that there is a duality that allows us to eliminate all elements with a depth greater than half the weight. When the depth is exactly half the weight we choose from a sum and its dual the element that comes first lexicografically.

We pull the function E outside brackets. The contents of a bracket is what we know about the object indicated by the indices of the function E. At first this is all trivial knowledge.
Assume now that we generate the stuffle relation

$$H_{0,1}H_{0,1} \;=\; H_{0,0,0,1} + 2H_{0,1,0,1}$$

The left hand side can be substituted from the tables for the lower weight MZVs. Hence it becomes ζ_2^2. The right hand side objects are replaced by the contents of the corresponding E brackets in the master expression. These are for now trivial substitutions. From the result we generate the substitution

```
id H(0,1,0,1) = z2^2/2-H(0,0,0,1)/2;
```

We apply this to the master expression. Hence the master expression becomes

```
FF =
   +E(0,0,0,1)*(H(0,0,0,1))
   +E(0,0,1,1)*(H(0,0,1,1))
   +E(0,1,0,1)*(z2^2/2-H(0,0,0,1)/2);
```

Let us now generate the corresponding shuffle relation:

$$H_{0,1}H_{0,1} \;=\; 4H_{0,0,1,1} + 2H_{0,1,0,1}$$

and replace the right hand side objects by the contents of the corresponding E bracket in the master expression. This gives

$$\zeta_2^2 \;=\; 4H_{0,0,1,1} + \zeta_2^2 - H_{0,0,0,1}$$

which leads to the substitution

```
id H(0,0,1,1) = H(0,0,0,1)/4;
```

We obtain

```
FF =
    +E(0,0,0,1)*(H(0,0,0,1))
    +E(0,0,1,1)*(H(0,0,0,1)/4)
    +E(0,1,0,1)*(z2^2/2-H(0,0,0,1)/2);
```

We also need the divergent shuffles and stuffles. This is done by including the shuffles involving the basic divergent object and breaking down the multiple divergent sums with the stuffle relations as in:

$$\begin{aligned} H_1 H_{0,0,1} &= 2H_{0,0,1,1} + H_{0,1,0,1} + H_{1,0,0,1} \\ &= -H_{0,0,0,1} + H_{0,0,1,1} + H_{0,1,0,1} + H_1 H_{0,0,1}; \end{aligned}$$

Substituting from the master expression we obtain the relation

$$0 = -\frac{5}{4} H_{0,0,0,1} + \frac{1}{2}\zeta_2^2$$

Hence the substitution

```
id H(0,0,0,1) = z2^2*2/5;
```

and finally the master expression becomes

```
FF =
   +E(0,0,0,1)*(z2^2*2/5)
   +E(0,0,1,1)*(z2^2/10)
   +E(0,1,0,1)*(z2^2*3/10);
```

Now we can read off the values of all MZVs of weight 4 that we went to compute. All other elements (dual or divergent) can be obtained from these by trivial operations that involve the use of one or two relations only.

In practise we are a bit more sophisticated. It is noticed that the master expression can become rather big and hence to make a single substitution on it for each equation gives much sorting overhead. Therefore we group the equations and first diagonalize this group as much as possible. Then we substitute the results of the entire group into the master expression. When the group has 1000 elements this would give 1000 substitution statements. The result would be 1000 pattern matchings per term. This is solved by using tables which FORM can use internally in a binary tree search. The result is a rather fast program.

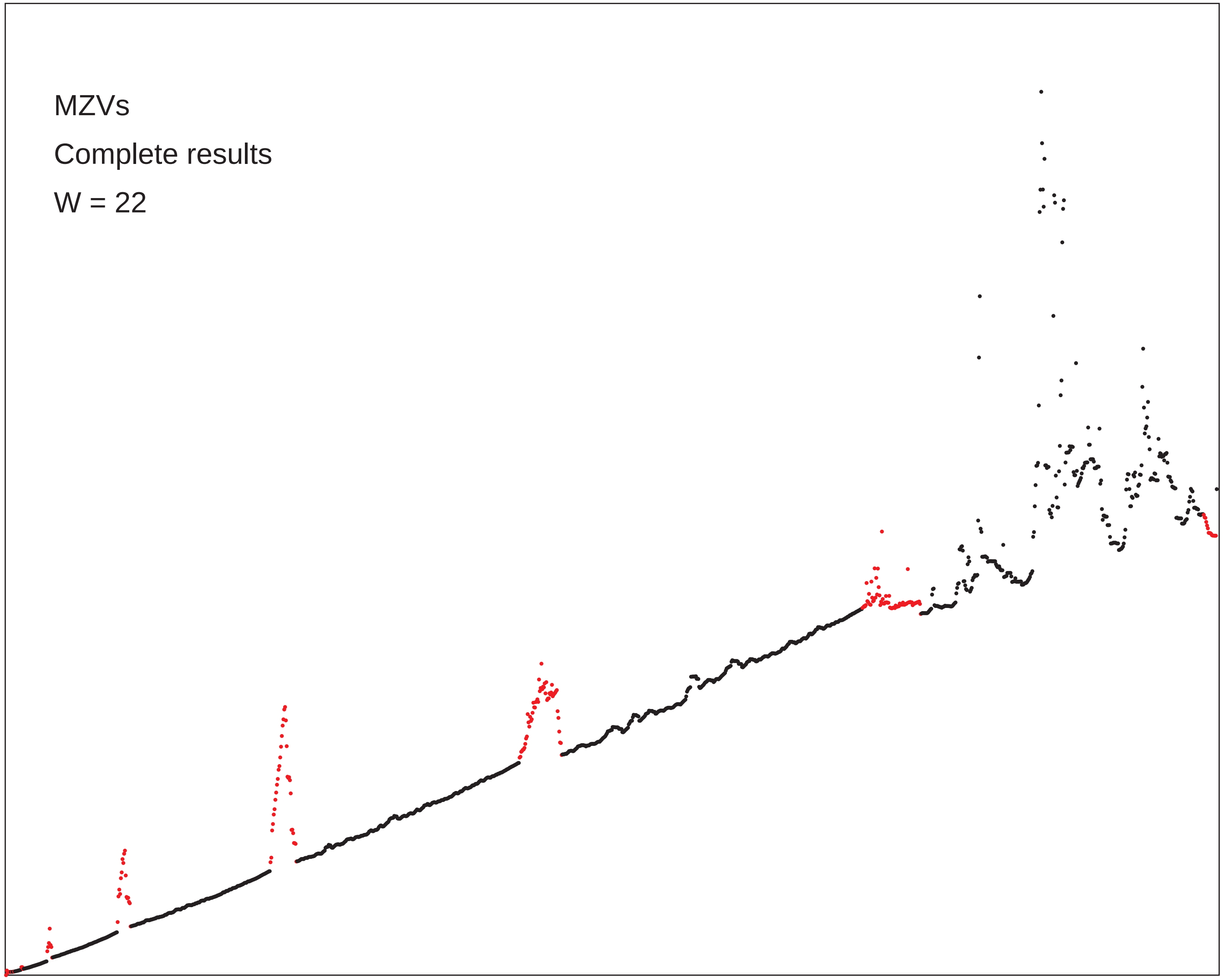
The size of the group is a function of the size of the problem. The optimal value is more or less related to the square root of the number of variables.

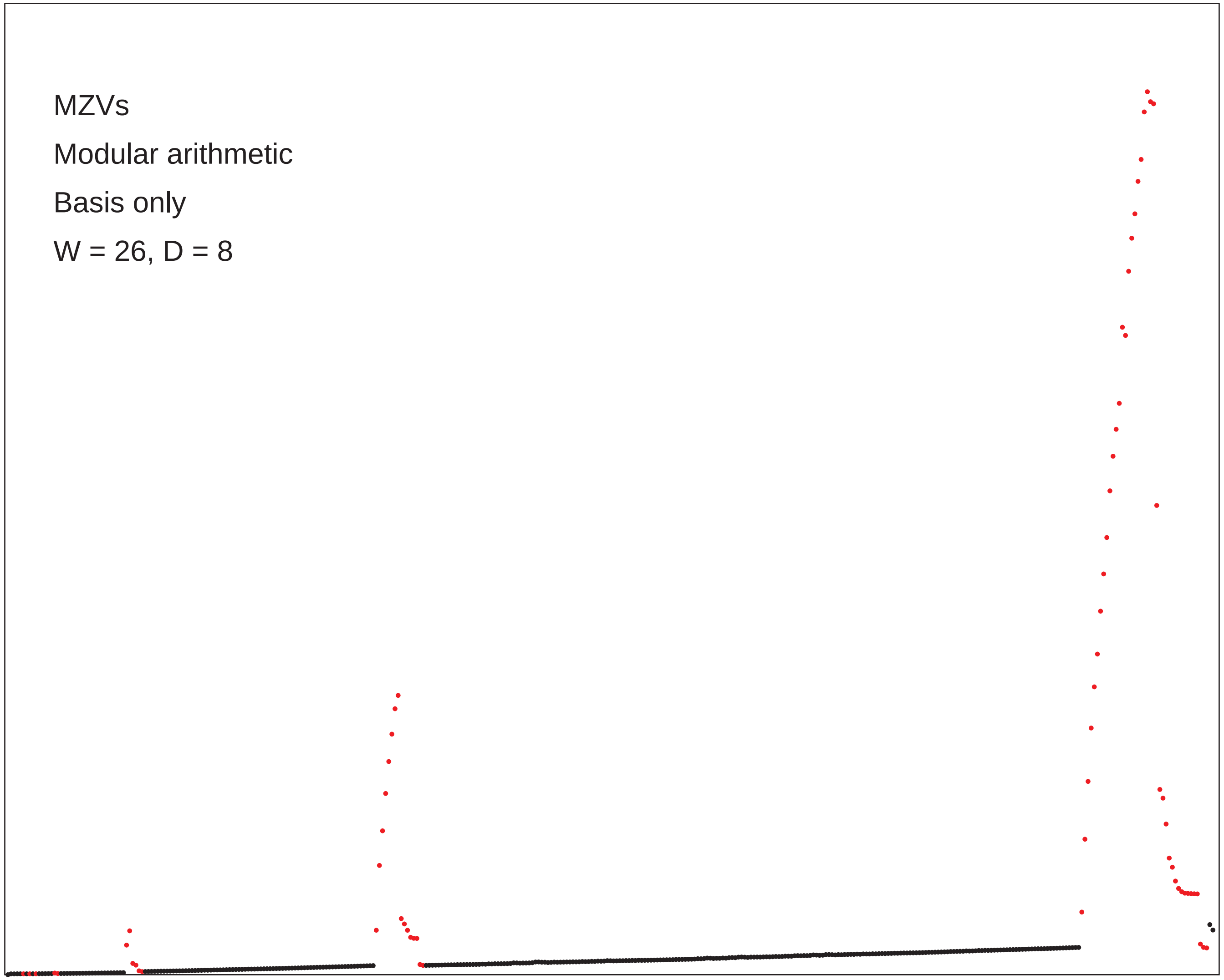
The whole program for the MZVs is only about 600 lines including commentary (400 lines in a stripped version). For the Euler sums it is a few hundred lines longer (the procedure for the generalized doubling relations is almost 200 lines).

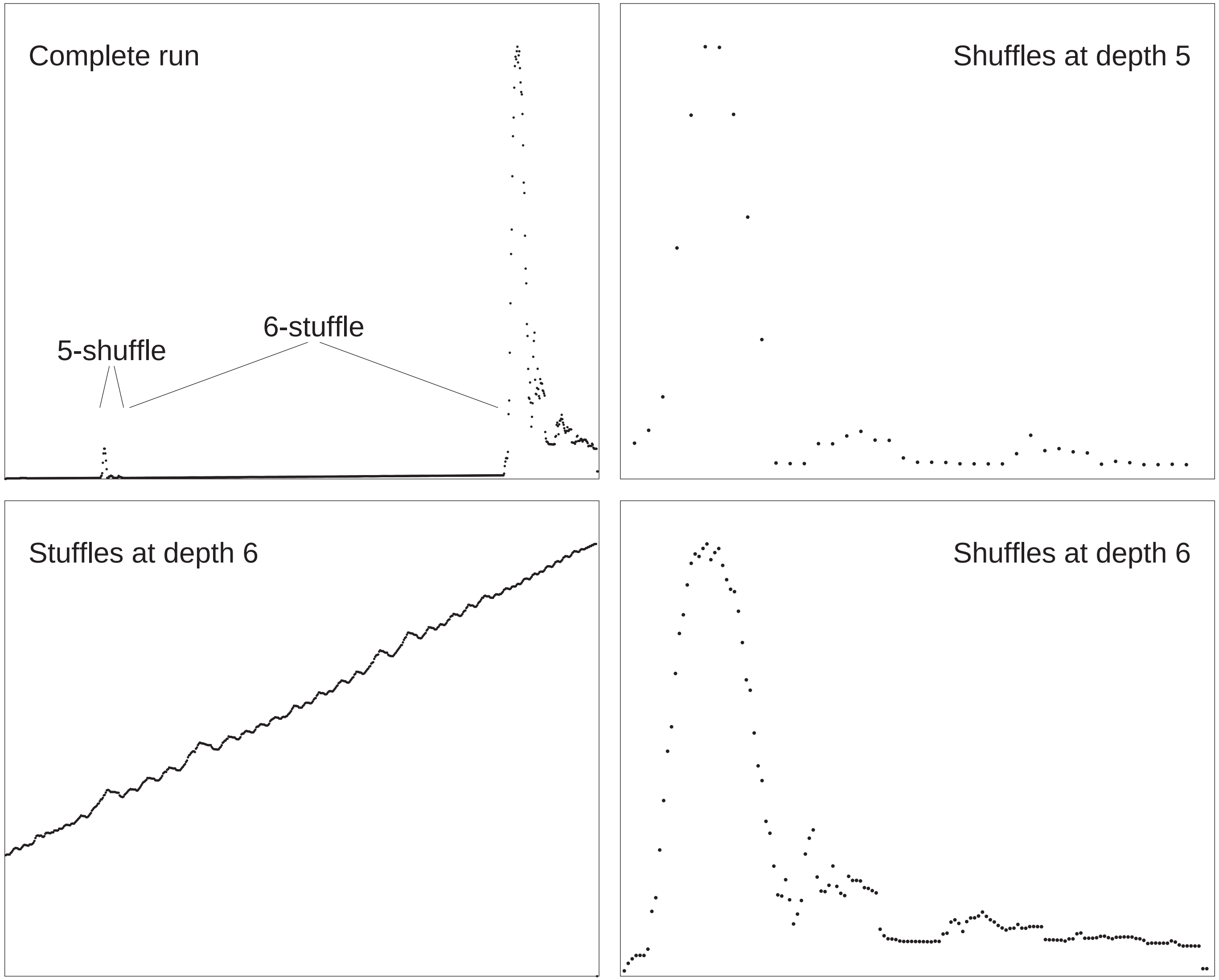
The major problem in the program is the order in which we feed in the equations. This can make a big difference in both the execution time and the space used (orders of magnitude!).

The stuffles don't cause too many troubles, but the shuffles are rather difficult to control. It is extremely hard to "block diagonalize" this system.

We have a heuristic ordering of the equations that we couldn't improve upon. Yet it still contains inefficiencies. This is shown in the following graphs in which we have on the x-axis the number of the module and on the y-axis the size of the output expression in that module.


MZVs
Complete results
W = 22


MZVs
Modular arithmetic
Basis only
W = 26, D = 8



Size of output expression for each module during phases of the Euler W=18,D=6 run.

We will run three types of programs.

1. A full expression of all MZVs in a minimal basis.
2. An expression of all MZVs in a minimal basis modulus a prime number. We drop all terms that are products of lower weight objects.
3. An expression of all MZVs in a minimal basis modulus a prime number. We drop all terms that are products of lower weight objects. We consider only elements up to a given depth D.

We run most of our programs on the computer of the theory group in Karlsruhe. This machine has 24 nodes, each node has 8 Xeon cores at 3 GHz with 32 Gbytes of memory and a 4 Tbyte disk. One of these nodes has been reserved for development work with TFORM and was hence used most of the time for this project. The other nodes were just running unrelated programs.

We also used some of the blade computers at DESY Zeuthen (8 Xeon cores at a somewhat lower frequency and 16 Gbytes of memory) and the main development machine for TFORM at Nikhef which has 4 Opterons at 2.3 GHz, 16 Gbytes of memory and a 1.5 Tbytes disk.

The last computer has also been used to compose the data mine.

.

Euler Sums

The Euler sums need the doubling ($W \geq 8$) and the generalized doubling ($W \geq 11$) formulas. They are also needed if we want to obtain results up to a given depth.

W	variables	eqns	remaining	size	output	time
4	36	57	1	4.3K	2.0K	0.06
5	108	192	2	21K	8.9K	0.12
6	324	665	2	98K	42K	0.37
7	972	2205	4	472K	219K	1.71
8	2916	7313	5	2.25M	1.15M	7.78
9	8748	23909	8	11M	6.3M	50
10	26244	77853	11	58M	36M	353
11	78732	251565	18	360M	213M	3266
12	236196	809177	25	3.1G	1.29G	47311

The size of the outputs becomes a bigger problem than the running time.

We have also runs with restricted depth. The most important ones are where we limit the depth to 6 or less. In this case we have used modular arithmetic and dropped all terms that are products of lower weight objects in an all out attempt to obtain $W = 18, D \leq 6$.

weight	constants	remaining	running time [sec]	output [Mbyte]
13	56940	22	2611	
14	90564	37	12716	51
15	138636	35	55204	87
16	205412	66	206951	214
17	295916	55	789540	288
18	416004	109	2622157	711

The last run was rather impressive. It took one month on an 8 core Xeon machine, working its way through a combined total of more than 7×10^{12} terms or 7 TeraTerms!

Runs to depth 5 are to weight 21 and runs to depth 4 are to weight 30.

MZVs

In the first sequence of programs we try to see how far we can get. We use a 31 bits prime (2147479273) and try to determine a basis. We drop all terms that are products of lower weight objects. We want expressions for all MZVs of the given weights in terms of the basis.

W	Group	size	output	CPU	time	Eff.
16	128	1.7M	1.2M	300	57	5.25
17	256	5.6M	3.2M	713	134	5.32
18	256	14.4M	7.2M	2706	465	5.82
19	512	39M	19M	6901	1206	5.72
20	512	104M	45M	30097	4819	6.25
21	1024	239M	114M	75302	12379	6.08
22	1024	767M	280M	449202	65644	6.84
23	2048	2.17G	734M	992431	151337	6.56
24	2048	8.04G	1.77G	9251325	1268247	7.29

At this point we noticed that all basis elements had a depth that fulfilled $D \leq W/3$. Hence assuming that this will be always the case we made a few more runs. And in addition we made some 'incomplete' runs.

W	D	size	output	CPU	real	Eff.
23	7	1.55G	89M	61447	9579	6.41
24	8	673M	380M	536921	72991	7.36
25	7	6.37G	244M	369961	50197	7.37
26	8	38.3G	1160M	4786841	651539	7.35
27	7	12.7G	914M	2152321	277135	7.77
28	6	2.88G	314M	235972	30960	7.62
29	7	41.0G	3007M	8580364	1112836	7.71
30	6	6.27G	658M	829701	106353	7.80

It shouldn't come as a great surprise that all the results of the above runs are in agreement with the Zagier and Broadhurst-Kreimer conjectures.

More later.....

We also made complete runs. That is: over the rationals and including products of lower weight objects. This gave the following:

W	size	output	num	CPU	real	Eff.	Rat.
16	10.9M	10.6M	21	254	59	4.29	1.05
17	30M	29M	19	690	149	4.62	1.11
18	86M	77M	25	3491	700	4.98	1.51
19	218M	205M	27	9460	1855	5.10	1.54
20	756M	552M	31	65640	11086	5.92	2.30
21	1.63G	1.55G	39	165561	27771	5.96	2.24
22	8.05G	4.00G	36	2276418	326489	6.97	4.97

It should become clear by now that the size of the output becomes a major obstacle. To store millions of expressions, each of them with quite a number of terms, will take Gigabytes.

```
Fill htable22(0,0,1,0,1,0,1,0,1,0,1,0,0,0,0,0,0,1,1,0,1,1)=229121/1728*
        z14z3z1z1z2z1+173609/576*z14z3z1z2z1z1+15692195/31104*
        z14z3z2z1z1z1+3726961/31104*z14z4z1z1z1z1-56339/1152*z14z5z1z2
        -3378973/13824*z14z5z2z1+1007419717/2488320*z14z6z1z1-3423/16*
        z15z2z1z2z1z1+2073365/1296*z15z3z1z1z1z1-307559/216*z15z4z1z2-
        666657535/165888*z15z4z2z1+2485272541/1658880*z15z5z1z1-
        502565387/31104*z16z2z1z1z1z1-8240323/1728*z16z3z1z2-
        50468588359/3317760*z16z3z2z1-4457267917/829440*z16z4z1z1-
        188177646093889/8599633920*z16z6+193151925403/19906560*
        z17z3z1z1+6998148491689/13271040*z18z2z1z1+5830492751924959/
        6879707136*z18z4-64399622164350811/1911029760*z20z2-1415173/
        43200*z5z3z3^2-141/4*z7z3*z8z2z1z1+15765715/62208*z7z3*z9z3+
        108/25*z5z3*z5z3z3z3+2332219/48600*z5z3*z9z5+654535363/5702400
        *z5z3*z11z3-30606548603/921600*z11*z5z3z3+4674331597474072633/
        57330892800*z11^2+3646960903267/217728000*z9*z5z5z3-
        26283756319/1451520*z9*z7z3z3+227618177777097021133/
        1504935936000*z9*z13-54161081/10368*z7*z10z2z1z1z1-14895806515/
        4644864*z7*z7z5z3+1810659173/497664*z7*z9z3z3+
        144492042468201625557/120394874880*z7*z15+7516571189/1126400*
        z7^2*z5z3-4571/5*z5*z5z3z3z3z3-27702313/5184*z5*z12z2z1z1z1+
        1897913010697639/388949299200*z5*z7z5z5-737558452534697/
        155579719680*z5*z7z7z3-8678023289443/13891046400*z5*z9z5z3+
        65728422985853/11112837120*z5*z11z3z3+185458251647/136857600*
        z5*z9*z5z3+655173768451/34836480*z5*z7*z7z3+
        898049408122901984292l/134420877803520*z5*z17-3819/4*z5^2*
        z8z2z1z1-271512575762737/20836569600*z5^2*z9z3-
        1538354691225468l/55564185600*z5^3*z7-2969/8*z3*z12z4z1z1z1+
        126/25*z3*z5z3z5z3z3-163253/400*z3*z5z5z3z3z3+5677/16*z3*
        z7z3z3z3z3-69740687/10368*z3*z14z2z1z1z1-374706432302269505/
        41015642443776*z3*z7z7z5+559257961960828567/109863327974400*z3
        *z9z5z5-675929428026804667/219726655948800*z3*z9z7z3+
        472645097440330207/97656291532800*z3*z11z5z3+17405218743810383/
        2048733388800*z3*z13z3z3+186/25*z3*z5z3*z5z3z3+560126822557/
        8294400*z3*z11*z5z3+241944929861/4976640*z3*z9*z7z3-48533/32*
        z3*z7*z8z2z1z1+3258424132907/44789760*z3*z7*z9z3+32205/16*z3*
        z5*z5z3z3z3+62730931353098707/4069012147200*z3*z5*z9z5-
        211693794294616819/4882814576640*z3*z5*z11z3-117303745103293/
        164229120*z3*z5*z7^2-3785404660891098517/439453311897 6*z3*z5^2
        *z9-979481944666231474286437l/109375046516736000*z3*z19-150567/
        1120*z3^2*z5z5z3z3+37369/224*z3^2*z7z3z3z3-76731/64*z3^2*
        z12z2z1z1+5836777489/4257792*z3^2*z11z5-631656298061/56609280*
        z3^2*z13z3-24/5*z3^2*z5z3^2-1476536914610227/4269957120*z3^2*
        z7*z9-940205/1728*z3^2*z5*z5z3z3-63798454917713/181149696*z3^2
        *z5*z11+89314457/907200*z3^3*z5z5z3-4391335/36288*z3^3*z7z3z3-
        102881298198157/1045094400*z3^3*z13+2015873/25920*z3^3*z5*z5z3
        +4771/112*z3^4*z7z3+178901285/1306368*z3^4*z5^2+129247787/
        466560*z3^5*z7-188/5*z2*z5z3z3z3z3z3-838*z2*z14z2z1z1z1z1-
        400090555909/130636800*z2*z7z3z5z5-860982225443/104509440*z2*
        z7z7z3z3-410971121201/87091200*z2*z7z5z5z3+432991955441/
        55987200*z2*z9z3z5z3-5561422085/1119744*z2*z9z5z3z3+
        30038614163/2488320*z2*z11z3z3z3+12317476820806379/11287019520
        *z2*z13z7-4814984387/46656*z2*z16z2z1z1-26973572103166541417/
        3386105856000*z2*z15z5+650628965993715945353/11512759910400*z2
        *z17z3+2703067/16128*z2*z7z3^2+15297217/51840*z2*z5z3*z9z3-
        1967338523/116640*z2*z9*z5z3z3+443971105937439694528 9/
        3837586636800*z2*z9*z11+203331234901/16329600*z2*z7*z5z5z3-
        2245163981/163296*z2*z7*z7z3z3+172861806934439936513/
        213199257600*z2*z7*z13-2530*z2*z5*z10z2z1z1z1+221934828641/
        37324800*z2*z5*z7z5z3-185137871143/18662400*z2*z5*z9z3z3+
        2356857770584504644547037/6120950685696000*z2*z5*z15-
        8784777689/466560*z2*z5*z7*z5z3-29339484871/12441600*z2*z5^2*
        z7z3-946617250799/97977600*z2*z5^4+4388/5*z2*z3*z5z3z3z3z3-
        2050*z2*z3*z12z2z1z1z1-2515919247697/1620622080*z2*z3*z7z5z5-
        5508608353973/1620622080*z2*z3*z7z7z3-65616653437/19293120*z2*
        z3*z9z5z3+4317757951/602910*z2*z3*z11z3z3+2459401/2880*z2*z3*
        z9*z5z3-5826659/2268*z2*z3*z7*z7z3+16858979284747836695237 33/
        19824227181158400*z2*z3*z17-3112*z2*z3*z5*z8z2z1z1-
        1913867931511/347276160*z2*z3*z5*z9z3-12126144556601/
        2083656960*z2*z3*z5^2*z7-1086/5*z2*z3^2*z5z3z3z3-4867384441/
        1088640*z2*z3^2*z9z5+71577340969/3991680*z2*z3^2*z11z3+
        11050634658317/143700480*z2*z3^2*z7^2+449759798507/4490640*z2*
        z3^2*z5*z9+128*z2*z3^3*z5z3z3-5793264895/139968*z2*z3^3*z11-
        207/5*z2*z3^4*z5z3-162/5*z2*z3^5*z5+27/5*z2^2*z12z2z1z1z1z1-
        984359/75600*z2^2*z7z5z5z1+2137981343/2721600*z2^2*z5z5z5z3-
        11370756889/1814400*z2^2*z7z5z3z3+1301016437/233280*z2^2*
        z9z3z3z3-7911180517/155520*z2^2*z14z2z1z1+336721679218271/
        4528742400*z2^2*z13z5-6306214666487812 9/62705664000*z2^2*z15z3
        +6644509/43200*z2^2*z5z3*z7z3-38514635023952878361/
        1630347264000*z2^2*z9^2-15429815879/1944000*z2^2*z7*z5z3z3-
        8274399031910863279/271724544000*z2^2*z7*z11+4208229059/544320
        *z2^2*z5*z5z5z3-658253387/77760*z2^2*z5*z7z3z3-
        8720289305450158267/952528896000*z2^2*z5*z13-50810851429/
        5443200*z2^2*z5^2*z5z3+999/5*z2^2*z3*z10z2z1z1z1-45306816419/
        2268000*z2^2*z3*z7z5z3+571783303/30375*z2^2*z3*z9z3z3+
        98747576355234045376281 7/127441460450304000*z2^2*z3*z15-
        670666193/72000*z2^2*z3*z7*z5z3-131835349/25920*z2^2*z3*z5*
        z7z3+73744749319/6531840*z2^2*z3*z5^3+1593/10*z2^2*z3^2*
        z8z2z1z1-5617847/40320*z2^2*z3^2*z9z3+113181386863/2177280*
        z2^2*z3^2*z5*z7+186543726721/6531840*z2^2*z3^3*z9-951/100*z2^2
        *z3^6+24711581/15120*z2^3*z5z5z3z3-234965329/136080*z2^3*
        z7z3z3z3-146515315/6048*z2^3*z12z2z1z1-435261786095987/
        7185024000*z2^3*z11z5+2456425078110467/7547904000*z2^3*z13z3+
        12415031/252000*z2^3*z5z3^2+117865176559161139/1046139494400*
        z2^3*z7*z9-226177577/45360*z2^3*z5*z5z3z3+
        5393961686980635863 69/212366317363200*z2^3*z5*z11+1568719081/
        661500*z2^3*z3*z5z5z3-811187497/317520*z2^3*z3*z7z3z3-
        26840936328970507766 81/953087845248000*z2^3*z3*z13-6731243/
        2800*z2^3*z3*z5*z5z3+1080509/15120*z2^3*z3^2*z7z3+2009725/168*
        z2^3*z3^2*z5^2+570093989/52920*z2^3*z3^3*z7+428519309/105000*
        z2^4*z5z3z3z3+20548647742626947/411505920000*z2^4*z9z5-
        910144972791054017/6035420160000*z2^4*z11z3-
        1373575155838415614 9/12070840320000*z2^4*z7^2-
        94688695713426099127/58342394880000*z2^4*z5*z9-141084539/78750
        *z2^4*z3*z5z3z3+14054410601686379371673 9/2601929817527040000*
        z2^4*z3*z11+17966741/252000*z2^4*z3^2*z5z3+5233954847/13608000
        *z2^4*z3^3*z5-89747783/12474*z2^5*z8z2z1z1+42587330003873/
        2235340800*z2^5*z9z3+197461454612336832 37/53480528640000*z2^5*
        z5*z7+1287323935999686801847583/30665601420854400 00*z2^5*z3*z9
        +1323224553841/1571724000*z2^5*z3^4+196664555715971051/
        22884301440000*z2^6*z7z3+689800062898138493 23/
        113556989145600 00*z2^6*z5^2+949714407130633561929828 73/
        104646364848665640000 0*z2^6*z3*z7+313619248788976309/
        44951306400000*z2^7*z5z3+909871564554223072 79/1064596773240000
        *z2^7*z3*z5+21641573024873924687/3863315055600000*z2^8*z3^2-
        288994255199496099205383627006427/162732217997457108000000 00*
        z2^11;
```

We have of course more results when we restrict the depth. They are less interesting from the viewpoint of this talk.

Results

The results of all our runs have been put in a location which we call the MZV datamine. It is more than 30 Gbytes of relations (after bzip2 compression) and it is located at

http://www.nikhef.nl/∼form/datamine/datamine.html

We will now show some of the results that we obtained from it.

The first things we look up in the datamine are some relations that Broadhurst discovered in the 1990's with the use of PSLQ. Now we can obtain 'formal' proof of them. They are so-called push down relations in which an object that has at least depth D as a MZV, can be expressed in terms of depth $D-2$ Euler sums.

The simplest example of such a push down relation is the following:

$$\begin{aligned} H_{6,4,1,1} = & -\frac{2107648}{15825}H_{-11,-1} + \frac{50048}{9495}H_{-9,-3} - \frac{117568}{237375}H_{-7,-5} \\ & +\frac{64}{243}H_{-3}^4 + \frac{69528448}{427275}H_{-3}H_{-9} + \frac{696654848}{4984875}H_{-5}H_{-7} \\ & +\frac{100352}{1583}\zeta_2 H_{-9,-1} - \frac{3584}{1583}\zeta_2 H_{-7,-3} - \frac{21236224}{299187}\zeta_2 H_{-7}H_{-3} \\ & -\frac{11690624}{356175}\zeta_2 H_{-5}^2 + \frac{320}{57}\zeta_2^2 H_{-7,-1} - \frac{64}{171}\zeta_2^2 H_{-5,-3} \\ & -\frac{11072}{1425}\zeta_2^2 H_{-5}H_{-3} - \frac{32}{35}\zeta_2^3 H_{-3}^2 - \frac{253512822078691 4}{481025690578125}\zeta_2^6 \end{aligned}$$

The next one at $W = 15$ becomes already rather bad.

$$
\begin{aligned}
H_{6,2,5,1,1} \;=\; & -\frac{28009182704961773376996398903118174942184754265798529122596}{30565191352147371108172627271581559533202207156609129 0625}\zeta_2^6 H_{-3} \\
& -\frac{68687238807894361714855018645761222083481069778506279 44}{38707190153725780323875000478239018538890298220085625}\zeta_2^5 H_{-5} \\
& -\frac{352620899448359235956708050628782983678844745342656}{1013638012410208225330902212029919741212540974465}\zeta_2^4 H_{-7} \\
& -\frac{450346189502746275947949624113680029363879966160832}{1079689612216387207432665263440390701792806802375}\zeta_2^3 H_{-9} \\
& +\frac{2176}{945}\zeta_2^3 H_{-3}^3 - \frac{2037950288768}{2234346324525}\zeta_2^2 H_{-3}^2 H_{-5} + \frac{176193784832}{29791284327}\zeta_2^2 H_{-3} H_{-7,-1} \\
& -\frac{195992987463712974831932122893210329859137445036 80}{4725229832288188719587664447056768718434401535 1}\zeta_2^2 H_{-11} \\
& -\frac{172882684928}{446869264905}\zeta_2^2 H_{-3} H_{-5,-3} - \frac{25300992}{8296097}\zeta_2^2 H_{-9,-1,-1} + \frac{74885120}{174218037}\zeta_2^2 H_{-7,-3,-1} \\
& +\frac{18508800}{58072679}\zeta_2^2 H_{-7,-1,-3} - \frac{111818752}{871090185}\zeta_2^2 H_{-5,-5,-1} - \frac{224668672}{7839811665}\zeta_2^2 H_{-5,-3,-3} \\
& -\frac{2212690676795201726 6176}{6122114344816491010 5}\zeta_2 H_{-3}^2 H_{-7} - \frac{3066450846132878467 6096}{4372938817726065007 5}\zeta_2 H_{-3} H_{-5}^2 \\
& +\frac{3632939862491022991 36}{3239213939056344 45}\zeta_2 H_{-3} H_{-9,-1} - \frac{4369910014768059392}{1079737979685448 15}\zeta_2 H_{-3} H_{-7,-3}
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1644070289092638208}{1841625107486235}\zeta_2 H_{-5} H_{-7,-1} - \frac{3361783780336378888}{5524875322458705}\zeta_2 H_{-5} H_{-5,-3} \\
&+\frac{853627469707858391615100678967489449812221696}{597132601687688036631228981023886538877725}\zeta_2 H_{-13} \\
&-\frac{58973326655000576}{40925002388583}\zeta_2 H_{-11,-1,-1} + \frac{117774300674867\,20}{122775007165749}\zeta_2 H_{-9,-3,-1} \\
&+\frac{204058184143646\,72}{613875035828745}\zeta_2 H_{-9,-1,-3} - \frac{8406294596950016}{613875035828745}\zeta_2 H_{-7,-5,-1} \\
&+\frac{1152979070087168}{368325021497247}\zeta_2 H_{-7,-3,-3} + \frac{12273867025183744}{613875035828745}\zeta_2 H_{-7,-1,-5} \\
&-\frac{2873606698310656}{1841625107486235}\zeta_2 H_{-5,-5,-3} - \frac{1792}{3645} H^5_{-3} \\
&-\frac{4256896288848871864427599757056}{345082792925864909648655961\,65} H^2_{-3} H_{-9} \\
&+\frac{3907508196189750773682657022328995\,84}{7122880994099869824756703677434\,25} H_{-3} H_{-5} H_{-7} \\
&-\frac{1208984451017729087145407744}{375907181836454149944069675} H_{-3} H_{-7,-5} \\
&+\frac{3840626217263581248362959360}{1353265854611234939798650\,83} H_{-3} H_{-9,-3} \\
&-\frac{40937844638235531233520436428\,8}{67663292730561746989932541\,5} H_{-3} H_{-11,-1} \\
&+\frac{224360652920825136173473713980416}{114217882882975498739996317\,3875} H^3_{-5}
\end{aligned}
$$

$$
\begin{aligned}
&-\frac{66613761278338041301228507648 0}{101527007007089332213330059 9}H_{-5}H_{-9,-1}\\
&+\frac{879380015176193352870400256}{376025951878108637827148 37}H_{-5}H_{-7,-3}\\
&-\frac{2844342500576392653874336768 0}{853006439162531976271187 49}H_{-7}H_{-7,-1}\\
&+\frac{568868500115278530774867353 6}{255901931748759592881356 247}H_{-7}H_{-5,-3}\\
&-\frac{2112533459510815147752919876950784}{15761057698646347406673907498 5}H_{-15}\\
&+\frac{852941656159907944394997 76}{712620249926927298472 17}H_{-13,-1,-1}\\
&-\frac{174907949900455846422691 84}{213786074978078189541651}H_{-11,-3,-1}-\frac{1258553193594283272003891 2}{213786074978078189541651}H_{-11,-1,-3}\\
&+\frac{467182771000149178765312 0}{213786074978078189541651}H_{-9,-5,-1}+\frac{487242448068471372021555 2}{1924074674802703705874859}H_{-9,-3,-3}\\
&+\frac{862712257577949234710528}{71262024992692729847217}H_{-9,-1,-5}-\frac{510117151499171079299072}{71262024992692729847217}H_{-7,-7,-1}\\
&-\frac{474464980999666928489984}{1924074674802703705874859}H_{-7,-5,-3}-\frac{247377046826432734064128}{641358224934234568624953}H_{-7,-3,-5}
\end{aligned}
$$

It just gives some more respect for Broadhurst who located these relations with the help of PSLQ in the 90's.

Verifying push downs isn't necessarily a trivial lookup in the tables. For example there are two MZVs at weight 17 and depth 5. There should be one push down. It is however a linear combination of the two that obtains the push down as in

$$H_{6,4,5,1,1} + \frac{72}{5} H_{5,3,3,3,3} \;\rightarrow\; (D \leq 3)$$

We do not show here the right hand side as it involves 99 terms. Just one term:

$$-\frac{391637561921020510388495527693101233498239312730472192870557734150375516560722487938377037680200651787224018403770884822305866731511488}{12096842033646879193852836812120840799898022503305835922565953025114624797521762549901601984894859006780341916995765114718351715625}\zeta_2^7 H_{-3}$$

This means that checks of the more complicated push downs require quite an amount of algebra first to get the 'non-push downs' out of the way.

Broadhurst and Kreimer gave a conjecture for the number of basis elements for each weight and depth for MZVs. They also gave a conjecture for each weight and depth when the MZVs are expressed in terms of Euler sums. These conjectures are given on the next page. In red are the numbers we explicitly verified.

From them one can see that there should be MZV basis elements that have fewer indices when expressed in terms of Euler basis elements as we have seen before. The push downs.

From the tables one can derive how many there should be, under the assumption that a push down is only from D to $D-2$.

W/D	1	2	3	4	5	6	7	8	9	10
1										
2	1									
3	1									
4										
5	1									
6		0								
7	1									
8		1								
9	1		0							
10		1								
11	1		1							
12		1		1						
13	1		2							
14		2		1						
15	1		2		1					
16		2		3						
17	1		4		2					
18		2		5		1				
19	1		5		5					
20		3		7		3				
21	1		6		9		1			
22		3		11		7				
23	1		8		15		4			
24		3		16		14		1		
25	1		10		23		11			
26		4		20		27		5		
27	1		11		36		23		2	
28		4		27		45		16		
29	1		14		50		48		7	
30		4		35		73		37		2

W/D	1	2	3	4	5	6	7	8	9	10
1										
2	1									
3	1									
4										
5	1									
6										
7	1									
8		1								
9	1									
10		1								
11	1		1							
12		2								
13	1		2							
14		2		1						
15	1		3							
16		3		2						
17	1		5		1					
18		3		5						
19	1		7		3					
20		4		8		1				
21	1		9		7					
22		4		14		3				
23	1		12		14		1			
24		5		20		9				
25	1		15		25		4			
26		5		30		20		1		
27	1		18		42		12			
28		6		40		42		4		
29	1		22		66		30		1	
30		6		55		75		15		

A push down basis

In determining a nice basis for the MZVs we noticed that the number of elements for each weight followed a prescription. They were equal to the number of elements one obtains when making all Lyndon words out of odd integers ≥ 3 in which the integers add up to the weight. Let us call this set L_W. The number of elements of a given weight and given depth in this construction follows exactly the second Broadhurst-Kreimer table! Next we tried to write as many basis elements as possible in terms of elements of this set.

This would not cover the whole set. The remaining elements could be obtained by allowing two even integers (say the first two indices) and making the last two indices equal to one. These elements would match the missing elements of our set if one would take away the ones and add them to the even integers. We call such a basis P_W.

Example: $W = 12$.

$$\begin{array}{lll} L_{12}: & H_{9,3} & H_{7,5} \\ P_{12}: & H_{9,3} & H_{6,4,1,1} \end{array}$$

Example: $W = 18$.

$$\begin{array}{lllllllll} L_{18}: & H_{15,3} & H_{13,5} & H_{11,7} & H_{9,3,3,3} & H_{7,5,3,3} & H_{7,3,5,3} & H_{7,3,3,5} & H_{5,5,5,3} \\ P_{18}: & H_{15,3} & H_{13,5} & H_{10,6,1,1} & H_{9,3,3,3} & H_{7,5,3,3} & H_{7,3,5,3} & H_{6,2,3,5,1,1} & H_{5,5,5,3} \end{array}$$

The interesting thing is that each of these special elements seems to be connected to a push down relation.

This is why we needed the run for Euler sums at $W = 18, D = 6$.

$$\begin{aligned} &H_{10,6,1,1}+46630979\ H_{5,5,5,3}+122713096\ H_{7,5,3,3}+1002156999\ H_{9,3,3,3} \\ &\longrightarrow 672686306\ H_{-17,-1}+72010179\ H_{-15,-3}-705663559\ H_{-13,-5} \\ &+817296192\ H_{-11,-7}+\cdots \end{aligned}$$

When we study the B-K tables and we study the outputs we have, we have to come to the conclusion that starting at weight 27 there should be a double pushdown.
i.e. We will have a basis element that needs 4 even indices and can be written with 4 trailing ones as in $H_{6,4,6,4,3,1,1,1,1}$ which would be expressed in terms of $H_{-7,-5,-7,-5,-3}$.
Unfortunately, the runs which would establish such a relation are extremely costly. Already the $W = 27, D \leq 9$ MZV run that would establish that we need the element with the four trailing ones is estimated to take of the order of one year. The corresponding run for the Euler sums to obtain the explicit pushdown relation would be much worse.
A similar but more mixed double pushdown should take place for $W = 28, D \leq 8$. This run seems 'cheaper'. It may take less than 3 months.

Currently running (since Aug-03): weight=28, depth $\leq$ 8).

This would be the first instance of a 'double pushdown'. Thus far (4-oct-2009):

```
TotalGeneratedTerms = 26195653989649;
```

which is 2.6 10^{13}. Statistics are like:

```
Time =    69738.22 sec    Generated terms = 6768912520814
               FF         Terms in output =    2563910243
substitution(8-sh)-4544 Bytes used      =   61564939480
```

The time is the time of the master. The 8 workers each contributed 2 weeks of CPU time in this step alone.

In total we managed to derive 16 explicit pushdown relations (up tp $W = 22$) using FORM and/or PSLQ. In them we noticed something interesting:
The part with the Euler sums could always be expressed as half of the right hand side of a doubling relation, provided we selected the basis correcty.
We introduce a new function A as

$$A_{n_1,n_2,\cdots,n_{p-1},n_p} = \sum_{\pm} sH_{\pm n_1,\pm n_2,\cdots,\pm n_{p-1},n_p}$$

in which the sum is over the 2^{p-1} possible sign combinations and $s = -1$ if the number of minus signs inside H is odd and $s = +1$ if this number is even as in

$$A_{7,5,3} = H_{7,5,3} - H_{-7,5,3} - H_{7,-5,3} + H_{-7,-5,3}$$

In terms of this function the pushdown relations we gave before become now

$$
\begin{aligned}
H_{6,4,1,1} \;=\; & -\frac{64}{27}A_{7,5}-\frac{7967}{1944}H_{9,3}+\frac{11431}{1296}H_5H_7-\frac{799}{72}H_3H_9 \\
& +\frac{1}{12}H_3^4+3\zeta_2H_{7,3}+\frac{7}{2}\zeta_2H_5^2+10\zeta_2H_3H_7 \\
& +\frac{3}{5}\zeta_2^2H_{5,3}-\frac{1}{5}\zeta_2^2H_3H_5-\frac{18}{35}\zeta_2^3H_3^2-\frac{5607853}{6081075}\zeta_2^6
\end{aligned}
$$

$$
\begin{aligned}
H_{6,4,1,1} \;=\; & -\frac{2107648}{15825}H_{-11,-1}+\frac{50048}{9495}H_{-9,-3}-\frac{117568}{237375}H_{-7,-5} \\
& +\frac{64}{243}H_{-3}^4+\frac{69528448}{427275}H_{-3}H_{-9}+\frac{696654848}{4984875}H_{-5}H_{-7} \\
& +\frac{100352}{1583}\zeta_2H_{-9,-1}-\frac{3584}{1583}\zeta_2H_{-7,-3}-\frac{21236224}{299187}\zeta_2H_{-7}H_{-3} \\
& -\frac{11690624}{356175}\zeta_2H_{-5}^2+\frac{320}{57}\zeta_2^2H_{-7,-1}-\frac{64}{171}\zeta_2^2H_{-5,-3} \\
& -\frac{11072}{1425}\zeta_2^2H_{-5}H_{-3}-\frac{32}{35}\zeta_2^3H_{-3}^2-\frac{2535128220786914}{481025690578125}\zeta_2^6
\end{aligned}
$$

$$
\begin{aligned}
H_{6,4,3,1,1} \;=\; & +\frac{1408}{81}A_{7,5,3}+\frac{16663}{11664}H_{9,3,3}+\frac{150481}{68040}H_{7,3,5}-\frac{20651486329}{4082400}H_{15} \\
& +\frac{1903}{120}H_7H_{5,3}-\frac{101437}{38880}H_5H_{7,3}-\frac{1520827}{38880}H_5^3+10H_3H_{6,4,1,1} \\
& +\frac{162823}{3888}H_3H_{9,3}-\frac{93619}{1296}H_3H_5H_7+\frac{3601}{48}H_3^2H_9-\frac{17}{20}H_3^5 \\
& +\frac{14}{5}\zeta_2H_{5,5,3}-2\zeta_2H_{7,3,3}+\frac{31753363}{12960}\zeta_2H_{13}-\frac{21}{2}\zeta_2H_5H_{5,3} \\
& -27\zeta_2H_3H_{7,3}-\frac{61}{2}\zeta_2H_3H_5^2-84\zeta_2H_3^2H_7-4\zeta_2^2H_{5,3,3} \\
& +\frac{979621}{1701}\zeta_2^2H_{11}-5\zeta_2^2H_3H_{5,3}+\frac{9}{2}\zeta_2^2H_3^2H_5-\frac{490670609}{3572100}\zeta_2^3H_9 \\
& +\frac{186}{35}\zeta_2^3H_3^3-\frac{1455253}{283500}\zeta_2^4H_7+\frac{4049341}{311850}\zeta_2^5H_5+\frac{12073102}{1488375}\zeta_2^6H_3
\end{aligned}
$$

What the above says is that we can find a good basis for the MZVs using the set L_w, provided we borrow some elements from the Euler sums. In such terms the basis for weight $\mathsf{w} = 18$ would look like

$$
\begin{array}{l}
L_{18}: Z_{15,3}\ Z_{13,5}\ Z_{11,7}\ Z_{9,3,3,3}\ Z_{7,5,3,3}\ Z_{7,3,5,3}\ Z_{7,3,3,5}\ Z_{5,5,5,3} \\
P_{18}: Z_{15,3}\ Z_{13,5}\ A_{11,7}\ Z_{9,3,3,3}\ A_{7,5,3,3}\ Z_{7,3,5,3}\ Z_{7,3,3,5}\ Z_{5,5,5,3}
\end{array}
$$

This is as close as we managed to come to a decent basis.

It is not unique which Z elements should be replaced by an A although the choice is much more restricted than for the first prescription (with the trailing ones).

The role of the doubling relation(s) here is also not quite clear. We can only study this for the weight 12 relations.

Conclusions

When calculating Feynman diagrams one runs into many interesting mathematical problems. Using the tools that were developed for Field Theory calculations we can often take the solutions way beyond what has been done before.
We have some conjectures about a basis for that MZVs that simplifies when they are embedded in the Euler sums. This may help the mathematicians in an attempt to obtain some solid theorems.
We did not obtain the holy grail of this field: a constructive algorithm that would express all MZVs or Euler sums in terms of a basis.
For now we will have to use table substitution when reducing MZVs and Euler sums to a minimal set.